

REMARKS

This is a Response to the Office Action mailed March 14, 2005, in which a three (3) month Shortened Statutory Period for Response has been set, due to expire June 14, 2005. Enclosed is our check to cover the fee for a one-month extension of time, to July 14, 2005. Claims 1-40 are pending.

Applicants thank the Examiner for noting that claims 10, 11, 16, 22, 23, 28, 29, 39, and 40 would be allowed if rewritten in independent form.

Rejections Under 35 U.S.C. § 103

Claims 1-3, 6, 8, 9, 12-15, 24-27, and 30-34 were rejected under 35 U.S.C. § 103(a) as being unpatentable over U.S. Patent No. 6,182,129 (Rowe et al.) in view of U.S. Patent No. 6,338,089 (Quinlan).

Claim 7 was rejected under 35 U.S.C. § 103(a) as being unpatentable over Rowe in view of Quinlan, and further in view of U.S. Patent Publication No. 2002/0038416 (Fotland et al.).

Claims 4, 5, 17-21, and 35-38 were rejected under 35 U.S.C. § 103(a) as being unpatentable over Rowe in view of Quinlan and further in view of U.S. Patent No. 6,014,702 (King et al.).

U.S. Patent No. 6,182,129 (hereinafter Rowe) appears to be directed to managing sessions with a host-based application using session vectors via link objects displayed on and selectable from Web pages. Rowe, Abstract. Rowe implements the session management in a system comprising three separate components: a remote computer, a host computer, and a server. Rowe, col. 6, lines 5-29, Figures 1-3. The session vector, maintained by a Session Vector Object instantiated at the remote computer, provides a mechanism for tracking sessions and linking a Web object (e.g., a field in a Web page) to a process resident at the remote computer. Rowe, col. 2, lines 38-48; col. 5, lines 58-63. Thus, access to a host-based session can be provided from a web browser by linking a user-selectable HTML field to an active session. Rowe, col. 2, lines 48-50. Rowe envisions removing both the session from the session vector and the link object from the Web page once a session has ended. Rowe, col. 9, lines 42-52, Figure 9.

U.S. Patent No. 6,338,089 (hereinafter Quinlan) appears to be directed to utilizing session pools for processing user generated requests via a standard browser on a client system to provide access to the facilities of server systems over a communication network. Quinlan col. 4, lines 11-25. As most clearly illustrated in Figure 1, Quinlan teaches locating the functionality including the session pool on the client system. Quinlan, col. 4, lines 15-25; col. 5, lines 48-57, Figure 1. The Quinlan system initiates the utilization of new session connections and the reuse of existing session connections as a function of requests issued from the browser. Quinlan, Abstract.

At client system startup, Quinlan appears to create a plurality of session pool table structures, defining an initial number of pre-established session connections. Quinlan, Abstract; col. 12, lines 42-47. Each subsequent session request may direct that a pre-established session connection can be obtained from a session pool table, in lieu of directing the system to open a new session connection. Quinlan, Abstract. When a session request signals that it requires a session but indicates that the session connection can be taken from a session pool, then a pre-existing session connection will be used if there is a session connection available in the pool; however, if there is no available entry in the session pool, then a new session connection is opened and placed into the session pool for immediate use. Quinlan col. 11, lines 7-38. A new session is created in this manner if the number of session pool entries generated is less than an established maximum value. Quinlan col. 11, lines 33-35. The initial pre-established session connections and new sessions created for the session connection pool will be returned to the pool for reuse when they are "logically closed," if the maximum number of session connections has not been exceeded. Quinlan, col. 12, lines 45-49.

U.S. Patent Publication No. 2002/0038416 (hereinafter Fotland) appears to be generally directed at enabling multithreading in an embedded processor, more specifically, a central processing unit (CPU). Fotland, Abstract; page 1, paragraph 0003. Fotland describes a system and method for enabling multithreaded processing in an embedded processor such that one thread may access and store information into the state of another thread. Fotland, page 1, paragraph 0012. Additionally, Fotland explains that multiple levels of priority are supported for

non real-time threads, and a low priority thread will give way to threads with higher priority. Fotland, page 5, paragraph 0064.

U.S. Patent No. 6,014,702 (hereinafter King) appears to be directed at providing programming capability for persistent, single implementation host access applications deployable across multiple platforms. King, Abstract; col 4, lines 16-20. The preferred embodiment is such that Java technology is used to implement the core host access function in a complete class model that is separable from other components (e.g. server platform), and a Java-enabled browser is required for operation. King, col. 4, lines 21-27. This appears to allow application developers to write Java applets that manipulate data from host applications without requiring that the applets reside on the users' machines. King, col 4, lines 28-35. These applets can open a session to a host, and, when interaction is complete, the applet can switch to other tasks or simply close the session. King, col. 4, lines 35-45; col 5, lines 15-21.

Each of the previously pending claims 1-9, 12-15, 17-21, 24-27, and 30-38 has been rejected as being unpatentable over Rowe in view of Quinlan, alone or in combination with either Fotland or King. However, each of the pending claims as previously presented includes features and functionality not disclosed by the cited prior art; therefore, each is allowable over these references.

In particular, pending independent claims 1 and 13 of the current application generally recite that a portion of available communication connections is maintained in one or more communication connection pools, such as by creating some number of available communication connections to be added to a pool of available communication connections when the portion of unused available connections is below a minimum value. For example, independent system claim 1 recites the following:

*"a communication connection pool configured to maintain in addition to communication connections ... being used ..., communication connections ... unused but available for use ...; a communication connection initiator configured to create the communication connections ...; [and] a communication connection pool manager configured to direct the communication connection initiator to create a first number of communication connections to be added to any unused available communication connections in the communication connection pool*

*when the number of unused available communication connections is below a second number.” (Emphasis added).*

This recited capability of maintaining a portion of unused available communication connections is useful, for example, for providing server-based applications with the ability to service anticipated client computers, which require access to legacy host computers, without having to incur the lag time generally associated with establishing a connection between the host computer and the server computer at the time of the client request.

Conversely, none of the cited prior art references appears to contain any teaching or suggestion of maintaining a portion of available communication connections in one or more communication connection pools, such as by creating some number of new available communication connections to be added to a pool when the quantity of unused available connections is below a number. For example, as previously discussed, Quinlan appears to describe a system that opens an initial number of “shared” session connections that are inserted into a session pool table at startup, Quinlan, col. 12, lines 42-47; thereafter, the “shared” sessions are provided for use upon request, but a new “shared” session connection is only created at the time that the client requests a session if there is no available entry in the session pool table, and the maximum value of sessions has not been exceeded. Quinlan col. 11, lines 28-38. As an illustrative example, if the initial session pool originally consisted of ten (10) session connections with a maximum value of one hundred (100), and those ten (10) sessions subsequently became used by ten (10) session requests, then there would be no available session left in the session pool table at that point (assuming that none of the requested sessions has been logically closed); therefore, the eleventh session request would cause a new shared session connection to be created for use by the requester, since there are no available connections in the pool, and the maximum value has not been exceeded. This technique of creating session connections one-by-one when there are no available connections in the session pool does not appear to teach or suggest Applicants’ claimed techniques of maintaining a portion of unused but available communication connections in anticipation of future client requests by adding to any unused available communication connections in the communication connection pool when the number of unused available communication connections is below another number.

Furthermore, as referenced above, Quinlan appears to explain that “shared” session connections are returned to the session connection pool for reuse after they are “logically closed,” but only if the maximum number of session connections has not been exceeded. Quinlan, col. 12, lines 45-49. However, this returning of existing connections to the session pool for reuse does not appear to teach or suggest Applicants’ claimed techniques for creating a number of additional communication connections to be added to the pool of available communication connections when the number of unused available communication connections is below a minimum value, such that a portion of available connections is maintained.

As pointed out by the Examiner in the previous Office Action, Rowe appears to provide no teaching of Applicants’ claimed communication connection pool (Office Action dated March 14, 2005, page 2, paragraph 3a); therefore, Rowe does not provide the teachings missing from Quinlan. Thus, for at least the aforementioned reasons, currently pending claims 1 and 13 are allowable over Rowe in light of Quinlan. Similarly, King does not appear to provide any teaching or suggestion to remedy the teachings missing from either Quinlan or Rowe. Thus, for at least the aforementioned reasons, claims 1 and 13 are patentable over the cited prior art references.

Similarly, currently pending claim 17 recites language that distinguishes it from the cited prior art references. For example, claim 17 recites the following:

“a screen object pool ... to contain available screen objects associated with communication connections ... to be available for use by the client computers ...; a ScreenFactory class configured to create the screen objects ...; [and] a screen pool manager configured to determine if the number of unused available screen objects is below a first number, and if so, the screen pool manager being configured to direct the ScreenFactory class to create a second number of screen objects to be *added to the unused available screen objects in the screen object pool.*” (Emphasis added).

None of the cited prior art references appears to contain any teaching or suggestion of a system that maintains a portion of available screen objects associated with communication connections to be available for use by client computers, wherein a screen pool manager is configured to determine if the number of unused available screen objects is below a

value and direct a ScreenFactory class to create a number of screen objects to be added to the unused available screen objects in the screen object pool when it is so determined. Thus, for at least this reason, currently pending claim 17 is patentable over the cited references.

Additionally, each of the currently pending method claims 24, 31, and 35 generally recite a method that maintains a pool of available communication connections by determining if the number of available communication connections is at least at a desired level greater than zero and increasing the number of available communication connections if the number is below the desired level. For example, claim 24 recites the following:

“maintaining a pool of available communication connections ... available for use by the client computers ... determining if the number of available communication connections in the pool ... is at least at a desired amount ... greater than zero; and increasing the number of available communication connections ... if the number of available communication connections ... is at or below the desired amount.”

None of the cited prior art references appears to contain any teaching or suggestion of maintaining a pool of available communication connections for use by client computers by determining that the available communication connections is below a desired value greater than zero and increasing the number of available communication connections upon that determination. For example, as previously discussed, Quinlan appears to create new “shared” session connections at the time that the client requests a session when there is no available entry in the session pool table and the maximum value of sessions has not been exceeded. Quinlan col. 11, lines 28-38. This technique of creating session connections one-by-one when there are no available connections in the session pool does not appear to teach or suggest Applicants’ claimed techniques of maintaining a pool of available communication connections in anticipation of client requests by determining that the number of available communication connections is below a desired value greater than zero and, if so, increasing the number of available connections. Also as pointed out previously, Rowe and King do not appear to provide any teaching or suggestion to remedy the teachings missing from Quinlan; therefore, for at least these reasons, currently pending claims 24, 31, and 35 are patentable over the cited prior art references.

The pending dependent claims include the features of those claims from which they depend, and are thus allowable for at least the same reasons as those claims. Furthermore, various pending dependent claims also recite additional features lacking in the cited prior art references and are allowable on the basis of those features as well. For example, claim 6 recites that “the pool manager is configured to apply operations research and queueing theory with historical traffic data of requests from the client computers ... to determine at least one of the first number and the second number.” Claims 14, 26, and 33 include similar language. Although Quinlan appears to discuss that the number of initial and maximum session connections are set from information included in a configuration file or received from a command line or administrative page (Quinlan, col. 12, lines 34-41), this does not appear to contain any teaching or suggestion to use historical traffic data of requests from clients to determine the values.

As another example, claim 9 recites that “the communication connection management system of claim 1 where in the first number is an increment.” Quinlan does not appear to teach a system where the first number is an increment as claimed by Applicants. As a further example, claim 19 recites that “the pool manager is configured to determine the first number and second number based in part upon levels of past requests from the client computers for access to the host computer through the server computer.” Claim 37 includes similar language. However, Quinlan appears to contain no teaching of a pool manager configured to determine the first and second number based in part upon levels of past requests from clients.

Other dependent claims also recite additional features lacking in the cited references, although these additional features are not enumerated here for the sake of brevity.

### Conclusion

Overall, the cited references do not singly, or in any motivated combination, teach or suggest the claimed features of the embodiments recited in independent claims 1, 13, 17, 24, 31, and 35, and thus such claims are allowable. Because the remaining claims depend from allowable independent claims 1, 13, 17, 24, 31, and 35, and also because they include additional limitations, these claims are likewise allowable. If the undersigned attorney has overlooked a

relevant teaching in any of the references, the Examiner is requested to point out specifically where such teaching may be found.

In light of the above amendments and remarks, Applicants respectfully submit that all pending claims are allowable. Applicants, therefore, respectfully request that the Examiner reconsider this application and timely allow all pending claims. Examiner Bayard is encouraged to contact Mr. Abramonte by telephone to discuss the above and any other distinctions between the claims and the applied references, if desired. If the Examiner notes any informalities in the claims, she is encouraged to contact Mr. Abramonte by telephone to expediently correct such informalities.

The Director is authorized to charge any additional fees due by way of this Amendment, or credit any overpayment, to our Deposit Account No. 19-1090.

Respectfully submitted,

SEED Intellectual Property Law Group PLLC

A handwritten signature in black ink, appearing to read 'Frank Abramonte', written over a horizontal line.

Frank Abramonte  
Registration No. 38,066

FXA:asl

Enclosure:  
Postcard

701 Fifth Avenue, Suite 6300  
Seattle, Washington 98104-7092  
Phone: (206) 622-4900  
Fax: (206) 682-6031

519539\_1.DOC